1 Check your understanding

1. What is the definition of an object language and a metalanguage?

2. Give several examples of situations where it is useful for the object and metalanguages to be different.

3. What is a quasiquotation? Give an example of quasiquotation from the paper and explain how it works.

4. What is an antiquotation? Give an example of antiquotation from the paper and explain how it works.

5. How is quasiquotation related to Template Haskell?

6. Explain what happens when the Haskell compiler encounters the `add` function defined in Figure 2. Who is responsible for defining the syntax that defines the antiquotation?

7. Explain what the function \( f \) in Figure 3 is doing. Explain what happens when function \( f \) is evaluated.

8. Explain how the function `peep` is evaluated.

9. Why is it important that quasiquotation parsers are passed line numbers?

10. Explain how the `subst` function on page 4 executes.

11. Explain what the `lame` and `lamp` functions do.

12. What is generic programming? (You may find the information on [http://www.haskell.org/haskellwiki/Generics](http://www.haskell.org/haskellwiki/Generics) useful.) What does it mean to extend a generic function with type-specific cases?

13. Explain how composing a generic function with `parse` yields a quasi-quoter. Explain what `dataToExpQ` and `dataToPatQ` do and how they differ.

14. Explain how the code in Figure 8 works. You may need to search online to find the definitions of `const` and `extQ`. (Try searching for `hackage extQ`, for example).

15. Explain why the type guarantees associated with quasiquotation can’t ensure generated the C code is type safe.
2 Evaluation Questions

1. What extra work must the user do when using quasiquotation as in this paper instead of quasiquotation as in the Template Haskell paper?

2. What are the advantages and disadvantages of implementing an Embedded DSL by providing a function compile that maps strings into host language code implementing the DSL program?

3. What are the advantages and disadvantages of implementing an Embedded DSL as a combinator library?

4. What are the advantages and disadvantages of implementing an Embedded DSL using quasiquotation vs. the approach taken in Slideshow?

5. What are the advantages and disadvantages of implementing a DSL using quasiquotation vs. developing a stand-alone DSL?

Useful reference:
Note that this paper describes the original design of GHC's Quasi-quotation library. It has evolved somewhat, and so you'll need to look at the online documentation to learn how to use the current version: http://www.haskell.org/haskellwiki/Quasiquotation